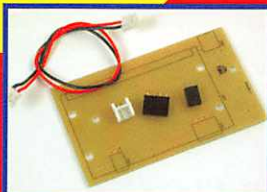
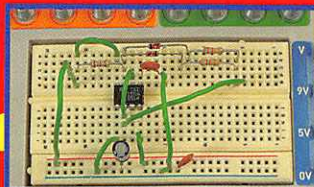
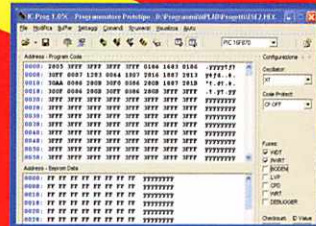
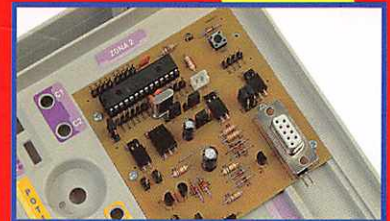
...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

36

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41036

Peruzzo & C.

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

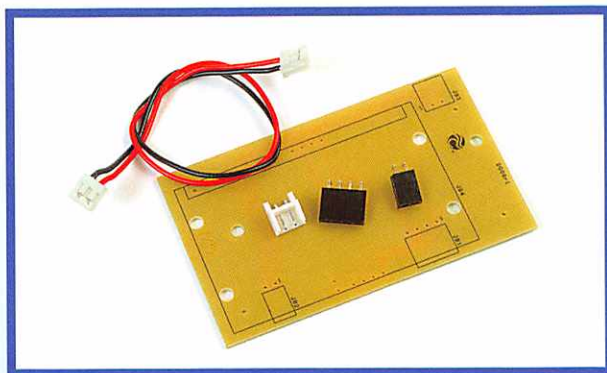
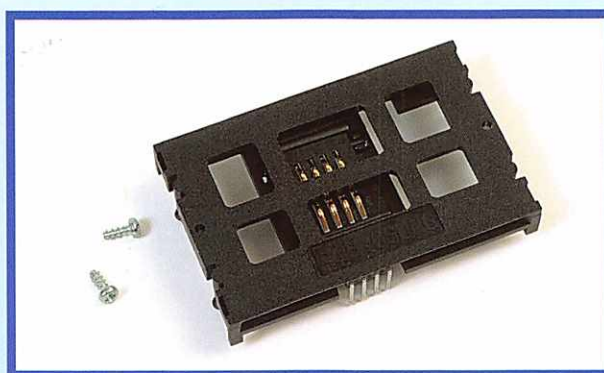
Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

impara l'elettronica digitale

IN REGALO in questo fascicolo

- 1 Scheda DG08r1
- 1 Cavetto a 2 fili con due connettori femmina a 2 vie
- 1 Connettore maschio da c.s. a 90° a 2 vie
- 1 Connettore femmina da c.s. a 90° a 2 vie
- 1 Connettore femmina da c.s. a 90° a 4 vie

IN REGALO nel prossimo fascicolo

- 1 Connettore da c.s. per SmartCard
- 2 Viti

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

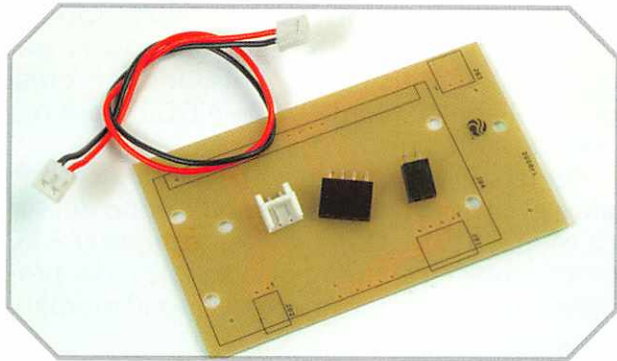
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

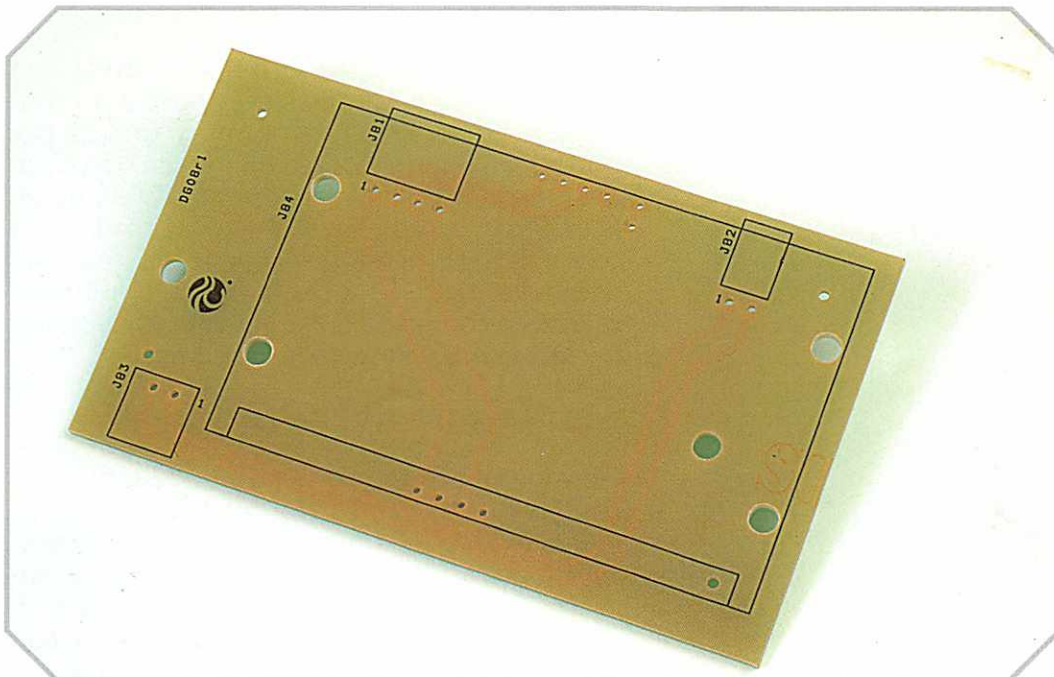
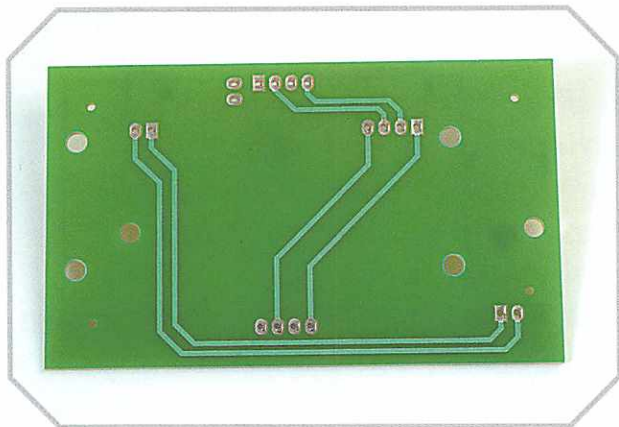
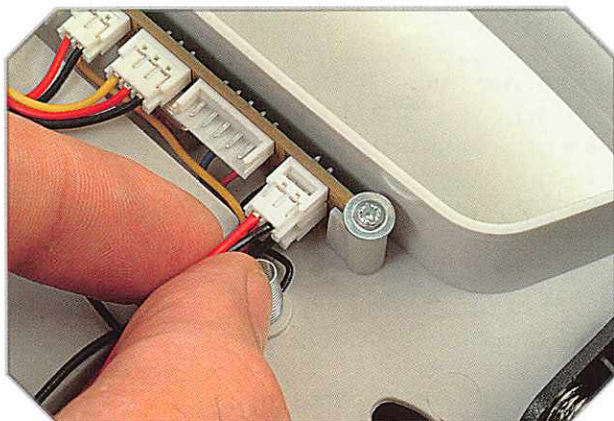
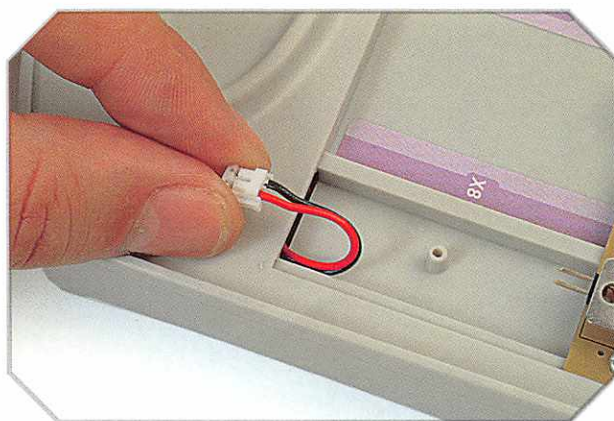
Lettura-scrittura di schede

Componenti forniti con questo fascicolo.

Cavetto per collegare l'alimentazione.

Con questo fascicolo viene fornito il circuito stampato che supporta il dispositivo di lettura-scrittura delle schede SmartCard o schede di memoria. Vengono forniti anche i connettori di questo circuito stampato e il cavetto di alimentazione. Monteremo il cavetto di alimentazione dei circuiti della zona 2 del laboratorio e inizieremo il montaggio della scheda DG08.

Lettura-scrittura

Il circuito stampato che ha come sigla DG08 supporta il dispositivo di collegamento delle schede SmartCard, necessario per poter eseguire il collegamento fisico alle schede di memoria.

Le schede DG08, DG07 e DG06 devono essere utilizzate insieme, in modo tale che il loro utilizzo, che denomineremo "normale", corrisponda alla situazione in cui il PIC funziona in modo autonomo senza collegamento al PC. Definiremo invece la posizione di scrittura, quella in cui esiste il collegamento al PC e in questa situazione inoltre, l'energia

Circuito stampato DG08, lato dei componenti

Circuito stampato DG08, lato della saldatura.

Collegamento all'alimentazione da 5 V, connettore J95 di DG09.

Il cavetto per ora non deve essere collegato.

per la scrittura è fornita dalla porta seriale.

Queste sono le situazioni generiche; per ogni caso specifico vi verranno indicati quali ponticelli delle schede DG06 e DG07 sarà necessario collegare.

Per leggere e scrivere le schede di memoria oltre al circuito DG08, avremo bisogno del circuito di scrittura DG07 e di un computer – PC o notebook – con una porta seriale e del programma apposito, come vedremo al momento opportuno.

L'alimentazione

Questo circuito stampato si utilizza per trasferire l'alimentazione da 5 V del laboratorio alla scheda DG07, e da questa alla scheda DG06, la quale ha realmente bisogno di questa alimentazione.

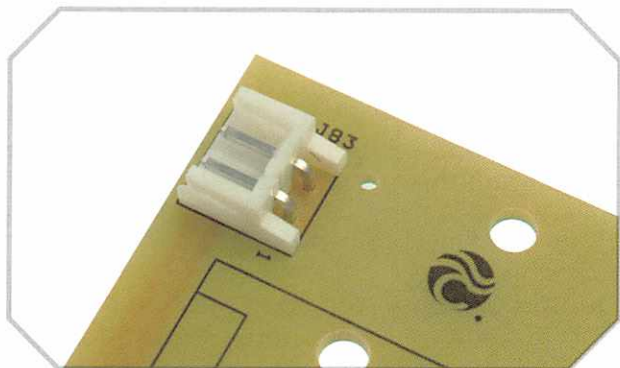
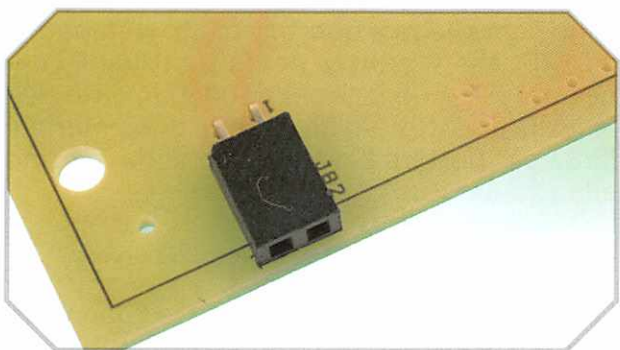
Le tensioni di scrittura si ottengono dalla porta seriale del computer a cui ci siamo collegati, per questo motivo dobbiamo fare molta attenzione al posizionamento dei ponticelli delle schede DG06 e DG07.

Guardando il circuito stampato dal lato delle piste, possiamo osservare che i collegamenti del connettore di ingresso dell'alimentazione J83, passano direttamente al connettore di uscita dell'alimentazione J82, senza essere usati da nessun altro circuito della scheda.

Il cavetto

Il cavetto a due fili, rosso e nero, fornito con questo fascicolo, ha alle sue estremità un connettore JST femmina a due vie. Si utilizza per portare l'alimentazione da 5 V dalla scheda di distribuzione dell'alimentazione DG09 fino alla DG08. Vi ricordiamo che quando si alimenta con le batterie in realtà si ottiene una tensione nominale di 4,5 V. Tuttavia i circuiti integrati utilizzati, compreso il PIC 16F870, funzionano perfettamente con questa tensione.

Montaggio del cavetto

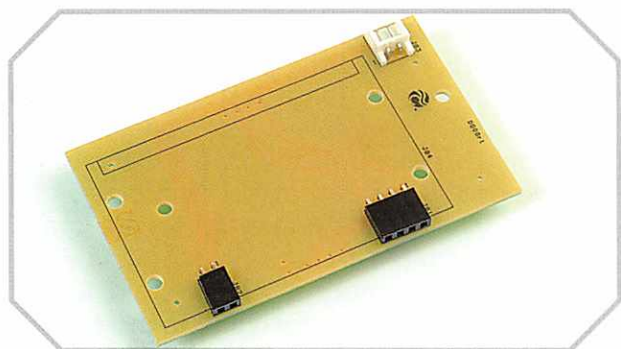
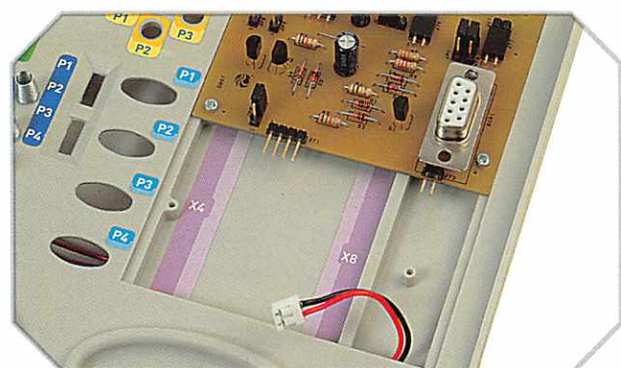
Per montare questo cavetto è necessario girare il pannello principale del laboratorio, in modo che i collegamenti della scheda DG09 risultino accessibili. Uno dei connettori del cavetto deve essere unito al connettore J95 di questa scheda, bisogna allinearlo bene con quello inserito sulla scheda, ed esercitare una

Connettore di ingresso dell'alimentazione.

Connettore di trasferimento dell'alimentazione a DG07.

pressione sufficiente per fare in modo che i due connettori risultino perfettamente incastrati e non si stacchino: potete osservare il connettore da uno dei lati per vedere il funzionamento del dispositivo di aggancio. Ricordate che questi connettori si rovinano se si cerca di scollegarli solamente con le mani, senza l'utilizzo di un cacciavite, come vi è stato indicato in precedenza. Una volta collegati è molto difficile che si stacchino in modo accidentale anche se lo strumento viene spostato, risulta sicuramente un vantaggio in strumenti portatili come questo, infatti i collegamenti saranno chiusi, quando vi verranno forniti i due coperchi che completeranno il laboratorio, permettendone il trasporto come una normalissima valigetta.

L'altro estremo del cavetto deve fuoriuscire dalla finestra, per il passaggio del cavo, che è situata nella parte inferiore della zona 2 del laboratorio. Questo cavetto deve essere lasciato libero sino a quando non sarà completata la scheda DG08 con tutti i suoi componenti.

Montaggio del circuito stampato

Il montaggio di questo circuito stampato inizierà con l'installazione del connettore di in-

Connettore per il dispositivo di collegamento della scheda di memoria.

Questo è l'aspetto attuale della scheda DG08.

In questa zona verrà montata la scheda DG08.

gresso dell'alimentazione, l'unico di colore bianco, che verrà inserito nella zona della scheda siglata J83 e con i terminali rivolti verso l'esterno della scheda. Prima di saldare i suoi terminali dobbiamo verificare che sia completamente appoggiata alla scheda.

Connettori J81 e J82

Questi due connettori sono del tipo a 90°, di tipo femmina e colore nero a quattro e due vie rispettivamente. Devono rimanere paralleli al circuito stampato e completamente appoggiati sullo stesso, fissandoli durante la fase di saldatura per evitare che si muovano e si possano inclinare. Questi connettori si utilizzano per collegare questo circuito stampato e il DG07, che si utilizza per la scrittura; il connettore J81 si utilizza per il collegamento della lettura-scrittura delle schede di memoria e il J82 per portare l'alimentazione da 5 V fino alla scheda DG07.

Montaggio della scheda

La scheda deve rimanere con i tre connettori montati in attesa che vi venga fornito il dispositivo di collegamento per la lettura-scrittura, per ora non dobbiamo collegare il cavo di alimentazione – connettore colore bianco – perché lo dovremmo comunque togliere per installare lo zoccolo della Smartcard, che vi sarà fornito con il prossimo fascicolo.

Laboratorio con il materiale montato fino a questo momento.

Astabile con 555

Abbiamo già studiato la parte teorica dell'integrato 555, realizziamo ora alcuni esperimenti con esso. In questa prova utilizzeremo lo schema base consigliato dal costruttore, però apporteremo una modifica per poter controllare in modo indipendente la durata del semiperiodo alto e basso dell'impulso, che si ripete in modo periodico.

Il circuito

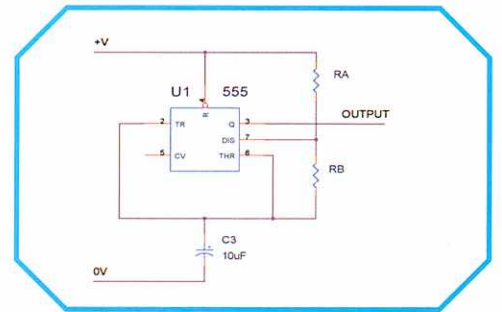
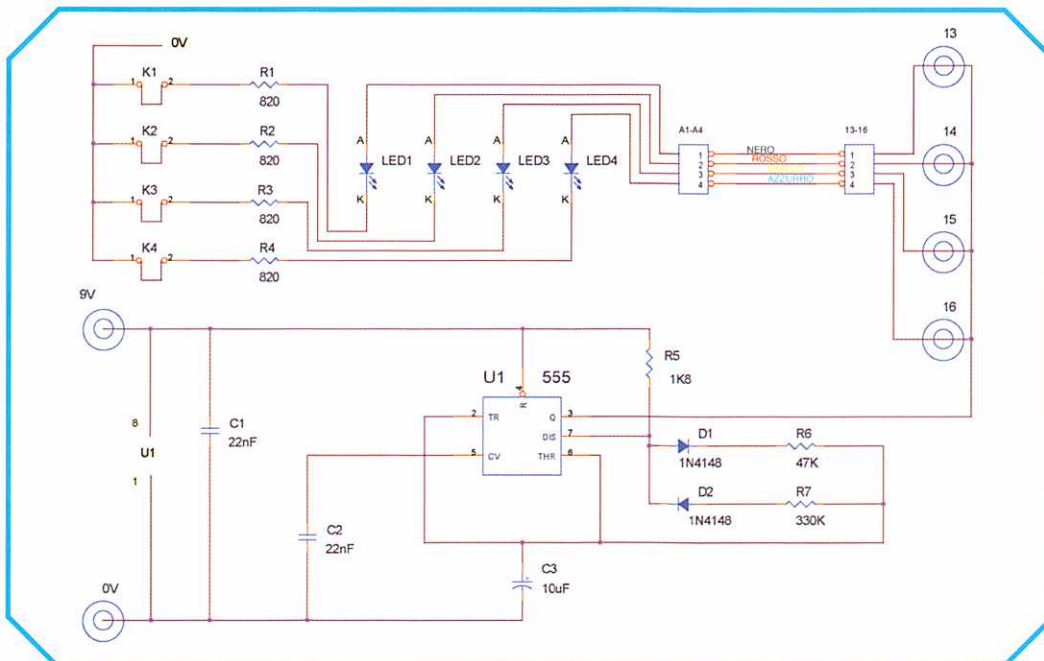
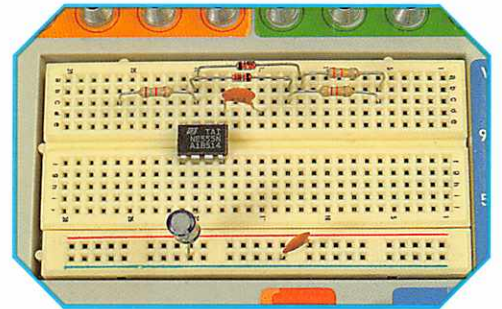
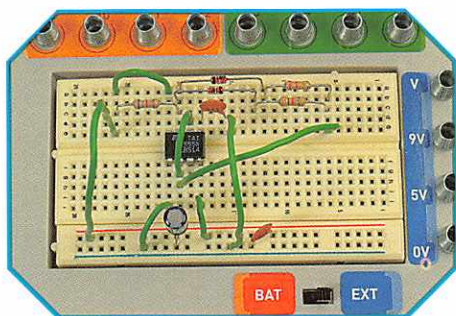
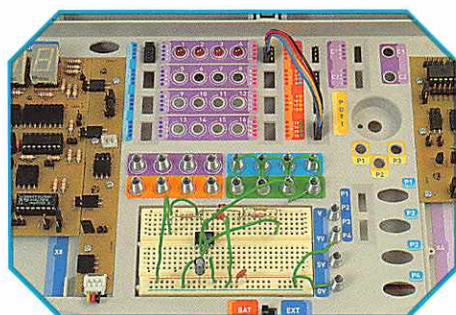
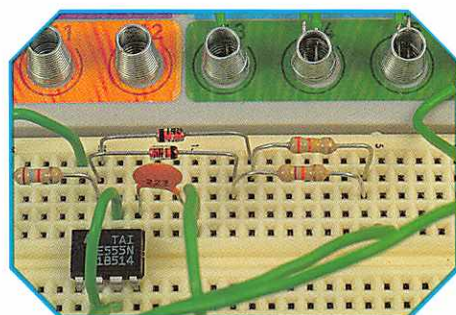
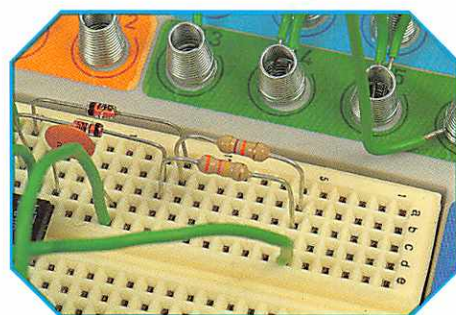
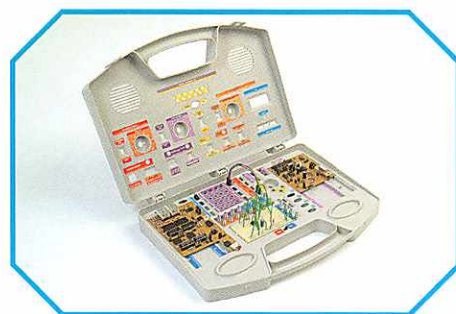
Se guardiamo lo schema del circuito completo e lo compariamo con il circuito di base, possiamo osservare che la resistenza RA del circuito di base si identifica come R5 sul circuito completo, per la resistenza RB invece, si utilizza la resistenza R6 per la carica del condensatore C3 e R7 per la scarica del medesimo condensatore. La separazione dei percorsi di carica e scarica del condensatore si ottiene con due diodi D1 e D2.

Il tempo in cui l'uscita permane a livello alto si calcola così:

$$T1 = 0,7 \times (R5 + R6) \times C3$$

E la permanenza a livello basso è:

$$T2 = 0,7 \times R7 \times C3$$

Schema di base.

Il 555 e altri componenti installati sulla scheda.

Astabile con 555, con controllo indipendente dei tempi di livello alto e di livello basso dell'impulso.

Cominciamo il cablaggio sulla scheda Bread Board.

Vista del cablaggio completo.

I diodi permettono di separare la carica e la scarica del condensatore.

Se si invertono le resistenze R6 e R7 si cambiano anche i tempi.

Esperimento completo.

È facile dedurre che se si desidera ottenere un segnale di uscita simmetrica, cioè che i due tempi T1 e T2 siano uguali, si deve compiere la seguente equazione:

$$R5 + R6 = R7$$

Bisogna ricordare che queste formule sono approssimative e devono essere verificate nella pratica, in quanto l'approssimazione della formula dipende dalle caratteristiche interne dell'integrato e dai valori dei componenti utilizzati, che non sono esatti a causa delle tolleranze accettate nella costruzione.

L'uscita del circuito, terminale 3, si utilizza per eccitare direttamente i quattro LED, dato che fornisce una corrente sufficiente.

Il condensatore C2 evita variazioni sull'ingresso di modulazione del circuito, rendendolo più stabile.

Montaggio

Questo montaggio si esegue seguendo lo schema, la lista dei componenti si ottiene dallo schema stesso.

I terminali di alimentazione dell'integrato del 555 sono: 1 per il negativo che si collega a 0 V e 8 per il positivo che si collega a 9V, ma si può anche alimentare a 5 V; il terminale di reset, terminale 4, deve essere collegato a livello alto, in modo da renderlo non operativo. Il condensatore C1 svolge funzione di filtro per l'alimentazione.

Infine, ricordate che il condensatore elettrolitico ha polarità, e che bisogna utilizzare un cavetto a quattro fili tra gli anodi dei quattro LED e i collegamenti dal 13 al 16 delle molle. Collocate i quattro ponticelli sui catodi dei LED.

Prova

Il circuito deve funzionare appena si collega l'alimentazione, i tempi di spegnimento e accensione dei LED dipendono dai valori delle resistenze R5, R6 e R7.

A titolo di prova vi consigliamo di utilizzare dei valori differenti, per iniziare potete scambiare fra loro i valori di R6 e R7.

Esercizio 2: programma combinazionale, la pratica

Per completare l'esercizio 2 dobbiamo fare il montaggio sul laboratorio. Questo montaggio sarà molto simile a quello eseguito per l'esercizio 1, però ci servirà per fare pratica e prendere familiarità con l'hardware. Prima di eseguire il montaggio finale del progetto, dobbiamo caricare il programma sul microcontroller e a questo scopo utilizzeremo IC-PROG. Infine, ci dedicheremo alla soluzione dell'esercizio 3, utilizzando lo stesso procedimento dei due precedenti.

Hardware per IC-PROG

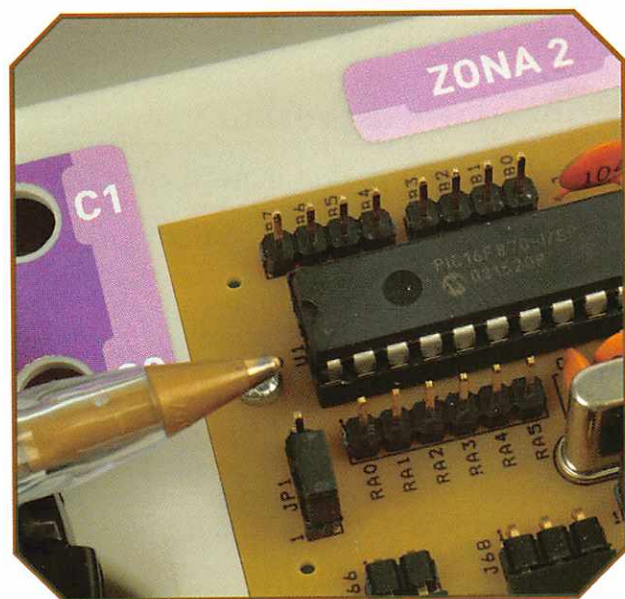
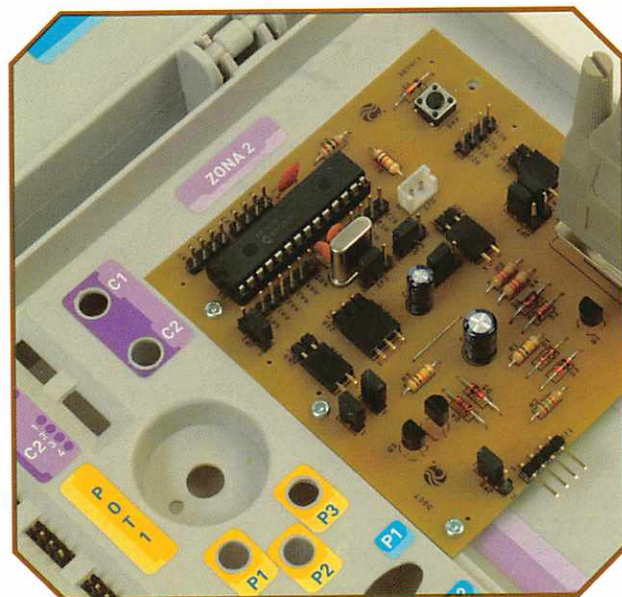
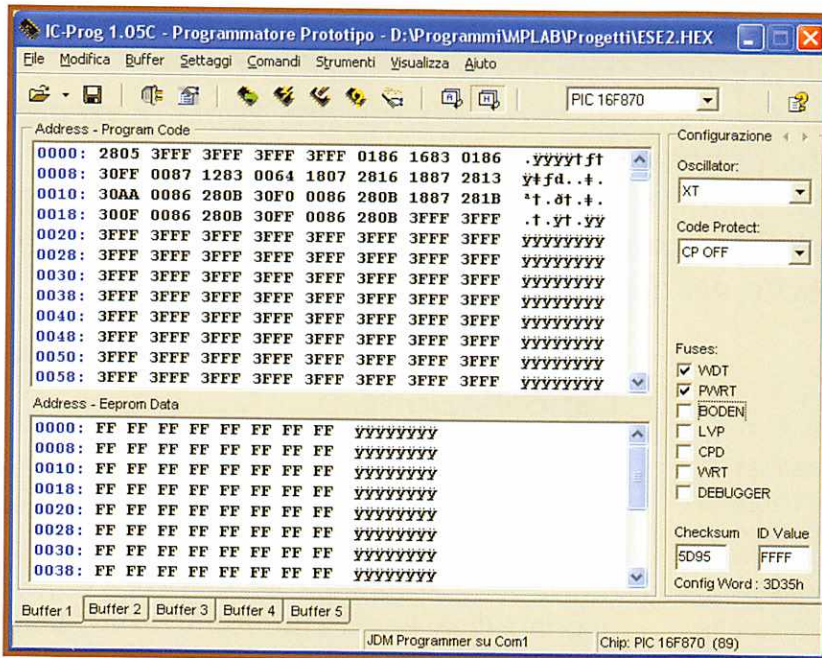
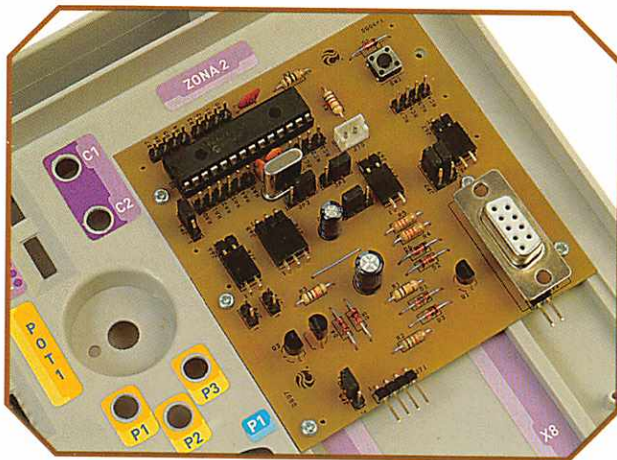
Supponendo che il PIC sia montato correttamente (inserito nella sua posizione corretta sullo zoccolo) configureremo i ponticelli del laboratorio. JP1, JP2 e JP3 dovranno avere i ponticelli fra i pin 1 e 2 per poter prendere l'alimentazione dal cavo che sarà collegato alla porta seriale del PC.

Dovremo inserire i ponticelli JP8 e JP9, che saranno stati tolti se avremo lavorato precedentemente in modo normale o di funzionamento, in altre parole se non avremo utilizzato il laboratorio per la scrittura del microprocessore.

Infine, collegheremo il cavo di comunicazione fra il laboratorio digitale e la porta seriale del PC.

Cancellazione

Dopo aver configurato l'hardware, siamo pronti per la programmazione del nostro dispositivo mediante IC-PROG. Come abbiamo già detto in altre occasioni è consigliabile cancellare il dispositivo che vogliamo scrivere prima di iniziare l'azione di scrittura. Leggeremo il dispositivo per verificare che non contenga programmi da salvare sul nostro PC, lo cancelleremo direttamente utilizzando il pulsante apposito presente sulla barra degli strumenti.

Alla richiesta di conferma di cancellazione, risponderemo affermativamente e verificheremo il risultato dell'operazione, leggendo nuovamente il dispositivo e controllando il contenuto delle posizioni di memoria.

Il PIC deve essere montato correttamente.

Configuriamo i ponticelli per il modo scrittura e colleghiamo il cavo di comunicazione.

Dopo aver caricato il programma IC-PROG dovrà avere questo aspetto.

Il laboratorio con il montaggio finale dell'esercizio.

Scrittura

Dobbiamo caricare il file che dobbiamo scrivere sul PIC, quindi selezioneremo Apri file, scegliendo il file desiderato. Dopo aver effettuato questo passaggio, la memoria di programma visualizzata sul software, cambierà di aspetto, dato che su essa si troverà il file in formato esadecimale che vogliamo scrivere.

Prima di eseguire il processo di scrittura dobbiamo configurare l'oscillatore e i bit della parola di configurazione. Selezioneremo un oscillatore di tipo XT, e cliccheremo i bit WDT e PWRT. Ricordate che non stiamo lavorando con un dispositivo dedicato a un programma specifico che non dovrà più essere modificato, quindi bisogna mantenere l'opzione di protezione del codice su CP OFF.

Quando tutto sarà pronto, potremo caricare il programma sul PIC, selezionando il pulsante Programma tutto o premendo il tasto F5. Dopo aver terminato il processo di scrittura, otterremo il messaggio che la verifica è stata effettuata in modo corretto. È consigliabile in ogni caso leggere nuovamente il dispositivo e assicurarci che il programma caricato, corrisponda a quello che volevamo scrivere.

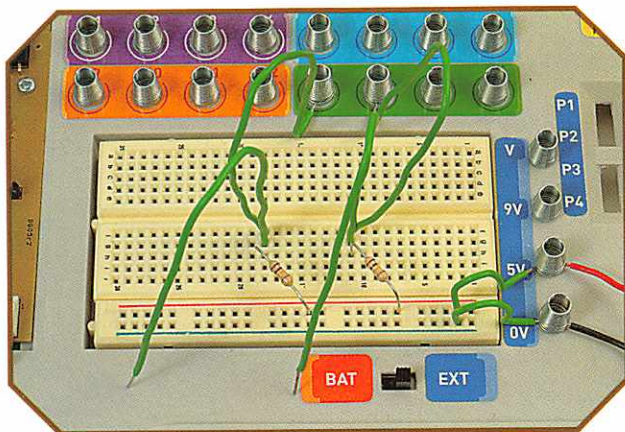
Configurazione hardware per il funzionamento

Scollegheremo il cavo di comunicazione e configureremo nuovamente i ponticelli del laboratorio per il modo di funzionamento normale. Cambieremo i ponticelli JP1, JP2 e JP3 in modo che ricevano l'alimentazione dalle batterie (pin 2 e 3 con ponticello) e toglieremo i ponticelli JP8 e JP9.

Montaggio: ingressi

Avremo bisogno di due ingressi che simuleranno gli interruttori e saranno collegati ai pin RC0 e RC1 del microcontroller. Come succedeva con l'esercizio 1, dato che non disponiamo di interruttori li simuleremo mediante due fili, che potranno essere collegati al positivo o al negativo dell'alimentazione. Come abbiamo fatto per l'esercizio precedente, utilizzeremo due resistenze, fra il pin di ingresso e il filo che andrà al positivo o al negativo. Collegheremo anche un cavetto di cui utilizzeremo solamente due fili a J72, per unire le alimentazioni.

Nell'immagine della figura possiamo osservare il montaggio degli ingressi e vedere che differisce dal montaggio dell'esercizio precedente solamente nel numero degli ingressi.

Collegamento degli ingressi sulla scheda Bread Board.

Collegamento degli ingressi sulla scheda DG06.

Montaggio: uscite

La porta di uscita è la porta B, e anche se la tabella della verità che utilizziamo per assegnare l'uscita contempla l'impiego di 8 bit, noi ne utilizzeremo solamente 4, dato che gli altri 4 saranno una ripetizione dei primi. Collegheremo mediante un cavetto i pin RB3:RB0 con gli anodi della matrice dei diodi e collegheremo i ponticelli sui catodi, come possiamo osservare nella figura.

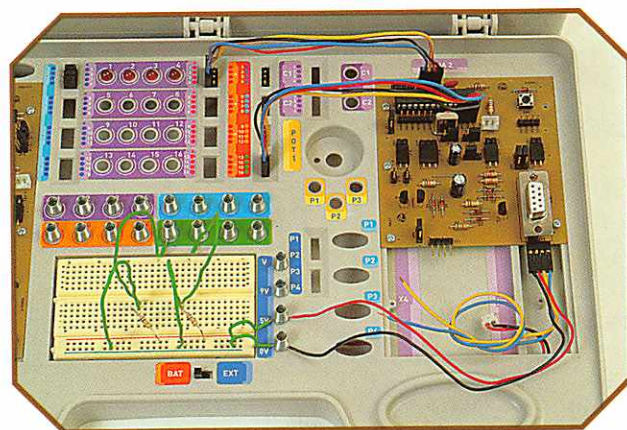
Funzionamento

Eseguito il montaggio completo (vedere immagine in basso) collegheremo l'alimentazione e osserveremo il funzionamento. Vedremo come cambiando lo stato degli ingressi, si otterranno combinazioni diverse sulle uscite. Supponiamo un sistema di illuminazione in cui disponiamo di due interruttori e di diverse zone da illuminare. Se il controllo lo realizziamo con un PIC, come previsto in questo esercizio, potremo scegliere combinazioni di zone in funzione dello stato dei due interruttori.

Come ultima cosa verificheremo che sul resto delle uscite otterremo lo stesso risultato. Se colleghiamo il cavetto della matrice dei LED ai pin di uscita RB7:RB4, come mostrato in figura, verificheremo che il risultato è lo stesso del precedente, perché le combinazioni delle uscite sono simmetriche.

Esercizio 3

Continuiamo con gli esercizi, facendo pratica con gli esempi, come l'esercizio 3 che è contenuto nel CD allegato all'opera, che vi proponiamo di risolvere. Si tratta di una variante dell'esercizio 1, in cui invece di visualizzare sull'uscita il valore dell'ingresso, viene mostrato il risultato della somma di quest'ultimo e di

Con un secondo cavetto collegheremo le uscite.

Montaggio completo dell'esercizio.

una costante. Cercate di risolvere voi stessi il programma e ricordate che se utilizzate la soluzione che si trova sul CD, questa non funzionerà, perché è stata pensata per essere scritta su di un PIC in cui è stato precedentemente caricato il programma residente Bootloader. Il montaggio di questo circuito è uguale a quello dell'esercizio 2, tenete presente però che abbiamo un ingresso nuovo (RC2), quindi non risulterà complicata la sua risoluzione hardware.

Per risolvere questo esercizio dobbiamo seguire gli stessi passaggi visti per i due esercizi precedenti: organigramma, codice, compilazione, simulazione, scrittura sul microcontroller e montaggio. Cambiate il valore della costante nel codice e verificate i risultati ottenuti, realizzate le modifiche pertinenti per la visualizzazione del risultato sul display a 7 segmenti invece che sui diodi LED. In altre parole, giocate con l'esercizio e fate tutta la pratica che potete, sia software che hardware, prendendo confidenza sia con la parte elettronica che con il microcontroller.

Ese3 - Blocco note

File Modifica Formato Visualizza ?

```
;
;Programma somma
;
;Al valore impostato tramite gli interruttori RC0-RC2 della porta C, si somma una
;constante (in questo esempio il valore 3). Il risultato si visualizza sui led RB0-RB7
;collegati alla porta B
;

        List    p=16F870            ;Processore
        include "P16F870.INC"       ;Definizione dei registri interni

        ORG     0xB0

Inizio  clrf    PORTB               ;Cancella i valori casuali
        bsf     STATUS,RP0          ;Seleziona il banco 1
        clrf    TRISB               ;Porta B si configura come uscita
        movlw   b'00000111'
        movwf   TRISC               ;3 bit della porta C si configurano come ingressi
        bcf     STATUS,RP0          ;Seleziona il banco 0

Loop:   clrwdt                      ;Aggiorna il WDT
        movf    PORTC,W             ;Carica lc stato degli ingressi RC0-RC2
        addlw   .3                  ;Somma 3
        movwf   PORTB               ;Visualizza il risultato su RB0-RB7
        goto    Loop

        end                         ;Fine del programma
```

Codice dell'esercizio 3 che si trova sul CD.

Verifichiamo che sugli altri pin di uscita il risultato è lo stesso.

Montaggio dell'esercizio 3 è uguale a quello dell'esercizio 2, ma con un ingresso in più, sulla scheda Bread Board.

Esercizio 4: temporizzatori, il programma

Continuiamo facendo pratica con gli esercizi e in questo caso utilizzando i temporizzatori per risolvere l'accensione sequenziale dei diodi LED. Realizziamo la programmazione e il montaggio come negli esercizi precedenti.

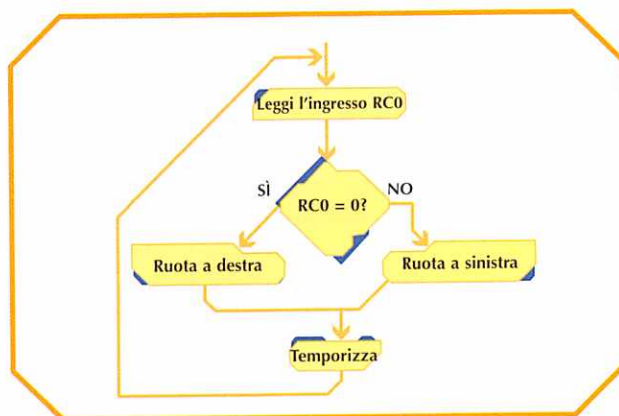
Enunciato

Si vuole realizzare una rotazione sequenziale nell'accensione di ogni LED collegato alla porta B. In base allo stato di un interruttore collegato al pin RC0 della porta C, la rotazione sarà da destra a sinistra (RC0 = 0) o in senso inverso (RC0 = 1). Ogni LED rimarrà acceso 250 ms.

Organigramma

Anche se in questo esercizio lavoriamo con i temporizzatori l'idea generale per risolverlo è molto semplice. Leggeremo l'ingresso e in funzione del valore di quest'ultimo sceglieremo una delle due possibili opzioni, ruotare a sinistra o ruotare a destra. In seguito eseguiremo la chiamata alla routine di temporizzazione e leggeremo nuovamente l'ingresso.

Per la routine di temporizzazione il programmatore normalmente utilizzerà la stessa routine creata per i progetti precedenti, nel nostro caso però supponiamo di non averla ancora sviluppata, quindi ne creeremo una.

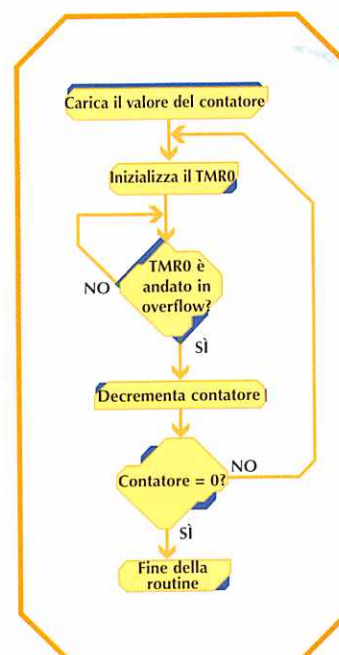
Nell'organigramma della figura, possiamo vedere la semplicità del progetto. Ogni volta che chiameremo la routine di temporizzazione caricheremo il valore del contatore, e inizializzeremo il temporizzatore. Terminata la temporizzazione decrementeremo il contatore e se non è arrivato a zero torneremo a inizializzare il temporizzatore ripetendo l'operazione. Quando il contatore arriva a zero, dichiareremo terminata la routine.

Organigramma generale dell'applicazione.

Codice

È molto importane cercare di risolvere gli esercizi da soli senza ricorrere alla soluzione che si trova sul CD. Inoltre come già succedeva per gli esercizi precedenti, la soluzione che si trova sul CD è pensata per essere scritta su un microcontroller che è stato precedentemente caricato con un programma residente.

La prima cosa che dobbiamo fare quando apriamo il Blocco note è scrivere mediante dei commenti ciò che vogliamo far eseguire al programma.

Dobbiamo utilizzare i commenti sia per la definizione generale di ciò che fa il programma che per definire la funzione di ogni istruzione utilizzata. Definiamo il tipo di processore e la libreria che contiene i registri, fatto questo dichiareremo le zone di memoria che dovranno contenere il programma con le direttive ORG.

Dobbiamo configurare i dispositivi, tenendo presente che ora ol-

Organigramma della routine di temporizzazione.

```
Inizio          clrf    PORTB           ;Cancella i valori casuali
                bsf     STATUS,RP0      ;Seleziona il banco 1
                clrf    TRISB           ;Porta B si configura come uscita
                movlw   0xFF
                movwf   TRISC           ;Porta C si configura come ingresso
                movlw   b'00000110'
                movwf   OPTION_REG      ;Prescaler da 128 per il TMR0
                bcf     STATUS,RP0      ;Seleziona il banco 0
```

Codice per configurare i dispositivi.

```
                bsf     STATUS,C        ;Attiva il carry
Loop            call    Delay           ;Temporizzazione da 250 ms
                btfsc   PORTC,0         ;RC0 = 0 ?
                goto    A_Destra        ;No, rotazione a destra
A_Sinistra      rlf     PORTB,F         ;Sì, rotazione a sinistra
                goto    Loop
A_Destra        rrf     PORTB,F         ;Rotazione a destra
                goto    Loop
```

Codice che risolve l'enunciato.

```
Delay           movlw   .10
                movwf   Contatore       ;Carica il contatore con 10
Delay_0         bcf     INTCON,T0IF     ;Azzera il flag di overflow del TMR0
                movlw   0x3c
                movwf   TMR0            ;Carica il TMR0
Delay_1         clrwdt                  ;Aggiorna il WDT
                btfss   INTCON,T0IF     ;Overflow del TMR0 ??
                goto    Delay_1         ;Non sono ancora passati i 25 ms
                decfsz  Contatore,F     ;Decrementa il contatore. E' stato ripetuto 10 volte ?
                goto    Delay_0         ;Ancora no, temporizza 25 ms
                return                  ;Ora sì
```

Routine di temporizzazione.

Ese4 - Blocco note

File Modifica Formato Visualizza ?

```
;Realizziamo una rotazione sequenziale nell'accensione di ogni LED collegato alla
;porta B. Se RC0 = 0, la rotazione sarà da destra verso sinistra e viceversa.
;Ogni LED rimane acceso 0.25 secondi (250 ms)

                List    p=16F870        ;Processore
                include "P16F870.INC"   ;Definizione dei registri interni

Contatore       equ     0x20            ;Variabile per la temporizzazione

                org     0x00
                goto    Inizio
                org     0x05

Inizio          clrf    PORTB           ;Cancella i valori casuali
                bsf     STATUS,RP0      ;Seleziona il banco 1
                clrf    TRISB           ;Porta B si configura come uscita
                movlw   0xFF
                movwf   TRISC           ;Porta C si configura come ingresso
                movlw   b'00000110'
                movwf   OPTION_REG      ;Prescaler da 128 per il TMR0
                bcf     STATUS,RP0      ;Seleziona il banco 0

                bsf     STATUS,C        ;Attiva il carry
Loop            call    Delay           ;Temporizzazione da 250 ms
                btfsc   PORTC,0         ;RC0 = 0?
                goto    A_Destra        ;No, rotazione a destra
A_Sinistra      rlf     PORTB,F         ;Sì, rotazione a sinistra
                goto    Loop
A_Destra        rrf     PORTB,F         ;Rotazione a destra
                goto    Loop

;*****************************************************************************
;Delay è una routine che realizza una temporizzazione da 250 ms, che è il tempo che deve
;Se il PIC lavora ad una frequenza di 4 MHz, il TMR0 conta ogni µs. Se si desidera tempo-
;rizzare 25000 µs (25 ms) con un prescaler da 128, il TMR0 dovrà contare 195 eventi
;(195 * 128 = 24960). Il valore 195 equivale a 0xc3 e, dato che il TMR0 è ascendente, lo dovremo
;caricare con il suo complemento a 2 (0x3c). Questa temporizzazione dovrà essere ripetuta 10 volte
;per ottenere il totale desiderato (250000)

Delay           movlw   .10
                movwf   Contatore       ;Carica il contatore con 10
Delay_0         bcf     INTCON,T0IF     ;Azzera il flag di overflow del TMR0
                movlw   0x3c
                movwf   TMR0            ;Carica il TMR0
Delay_1         clrwdt                  ;Aggiorna il WDT
                btfss   INTCON,T0IF     ;Overflow del TMR0?
                goto    Delay_1         ;Non sono ancora passati i 25 ms
                decfsz  Contatore,F     ;Decrementa il contatore. E' stato ripetuto 10 volte?
                goto    Delay_0         ;Ancora no, temporizza 25 ms
                return                  ;Ora sì

                end                     ;Fine del programma
```

Codice finale dell'applicazione.

tre agli ingressi e alle uscite, utilizzeremo un temporizzatore. Definiamo la porta B come uscita e la porta C come ingresso e sul registro delle opzioni OPTION_REG inseriremo un predivisore di valore 128.

Il programma lo risolveremo utilizzando le istruzioni di rotazione, quindi prima dovremo attivare il Carry, bit che si trova sul registro STATUS. A questo punto inizia l'anello in cui si chiama la routine di temporizzazione, si legge l'ingresso e mediante un salto condizionale discriminiamo i due possibili stati. Ruotiamo in un senso o nell'altro in funzione del valore dell'ingresso e mediante un salto incondizionale ripetiamo il giro. Nell'immagine della figura potete vedere la semplicità con cui abbiamo risolto l'applicazione.

Salva il file nella directory di lavoro di MPLAB.

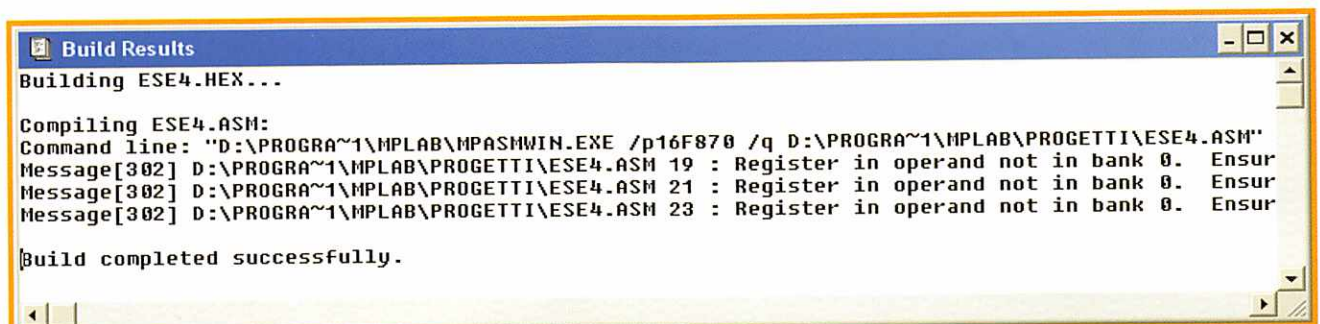
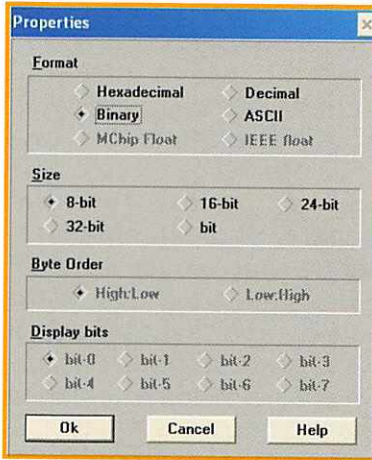
Definiremo una costante all'inizio del programma che utilizzeremo come contatore e inizializzeremo la routine caricando su di essa il valore del conteggio, che nel nostro caso è 10. Inizieremo il ciclo di temporizzazione cancellando il flag di overflow del Timer, dato che ogni volta che si ripete il ciclo è perché il timer ha terminato il suo conteggio e ha attivato questo flag. Caricheremo il temporizzatore con il valore del conteggio e attenderemo in un ciclo che il temporizzatore termini di contare. Quando questo avviene, esso attiva un flag per avvisarci dell'evento, a questo punto decrementeremo il contatore. Nel caso in cui il contatore raggiunga il valore zero, termineremo la routine, in caso contrario la ripeteremo. Nella terza figura della pagina precedente possiamo vedere il codice relativo a questa routine.

Routine di temporizzazione

Per realizzare la routine di temporizzazione dobbiamo fare prima dei calcoli. Se il PIC lavora a una frequenza di 4 MHz, il TMR0 evolve ogni microsecondo. Se si desidera temporizzare 25.000 µs, cioè 25 ms, con un predivisore di 128, il TMR0 dovrà contare 195 eventi (195x128 = 24.960). Il valore 195 inoltre equivale a 0xc3, dato che il TMR0 è ascendente, sarà necessario caricare il suo complemento (0x3c). Questa temporizzazione dovrà essere ripetuta 10 volte per ottenere il totale desiderato (250.000 µs o 250 ms).

Compilazione

Completato il codice lo dovremo compilare per verificare che non abbia errori. Ricordate che si deve salvare il codice con estensione ".asm" e selezionando nel campo Tipo l'opzione "Tutti i file", all'interno della cartella di lavoro che utilizziamo in MPLAB.

Passeremo alla compilazione del file, creando un progetto, includendo in esso il nostro fi-

Risultato della compilazione.

Visualizzazione dei registri in binario.

```
Delay           movlw   .10
                movwf   Contatore       ;Carica il contatore con 10
;Delay_0                bcf     INTCON,T0IF     ;Azzera il flag di overflow del TMR0
;               movlw   0x3c
;               movwf   TMR0            ;Carica il TMR0
;Delay_1                clrwdt                  ;Aggiorna il WDT
;               btfss   INTCON,T0IF     ;Overflow del TMR0?
;               goto    Delay_1         ;Non sono ancora passati i 25 mS
;               decfsz  Contatore,F     ;Decrementa il contadore. E' stato ripetuto 10 volte?
;               goto    Delay_0         ;Ancora no, temporizza 25 mS
                return                  ;Ora sì

                end                     ;Fine del programma
```

Modifica della routine di temporizzazione trasformando le linee di codice in commenti.

le, e selezionando l'opzione Build All. Nella figura in basso possiamo verificare il risultato ottenuto dalla compilazione.

Simulazione

Per simulare questo programma dobbiamo aprire la finestra dei registri delle funzioni speciali e crearne una in cui potremo vedere solamente i registri delle porte. In questa finestra, per rendere più semplice la visualizzazione, definiremo la rappresentazione dei registri in modo binario. Cliccando con il pulsante sinistro del mouse l'angolo superiore sinistro della finestra, si aprirà un menù in cui selezioneremo Edit Watch. Fatto questo apparirà una nuova finestra in cui per ogni registro selezioneremo Properties, cambiando in binario la rappresentazione. Avremo bisogno anche del simulatore di stimoli asincroni per verificare che, cambiando lo stato degli ingressi, l'accensione sequenziale dei LED venga eseguita in senso contrario. Configurate unicamente il pin RC0 su questo simulatore.

Per evitare di rimanere bloccati all'interno della routine di temporizzazione durante la simulazione, modificheremo il programma e trasformeremo in commenti la maggior parte della routine di temporizzazione, evitando che vengano eseguite le istruzioni che potrebbero provocare il blocco. Nell'immagine della figura in alto possiamo vedere i cambiamenti eseguiti. Salvate il programma e compilate nuovamente.

Se simuliamo il codice in modo passo a passo mediante F7, potremo verificare che sul registro di uscita il bit attivato esegue la rotazione. Verificheremo che il funzionamento della rotazione sia quello desiderato.

Prima di dichiarare terminata la simulazione dobbiamo riportare allo stato originale la routine di temporizzazione eliminando i ";" inseriti precedentemente, e compilare nuovamente il programma per ottenere il file desiderato in codice macchina.

Il montaggio hardware di questo progetto corrisponde al montaggio degli esercizi precedenti. Provate a eseguire la scrittura del programma sul PIC e il montaggio hardware senza ricorrere ai suddetti esercizi precedenti.

Aspetto di MPLAB.